International Journal of Electrical and
Electronics Engineering Research (IJEEER)
ISSN(P): 2250-155X; ISSN(E): 2278-943X
Vol. 6, Issue 6, Dec 2016, 37-46

COMPARISON BETWEEN PARAMETER OPTIMAL AND PID TYPE ILC FOR ONE-LINK ROBOT MANIPULATOR

MUTITA SONGJUN

Research Scholar, Department of Electrical and Computer Engineering, Naresuan University, Thailand

ABSTRACT

A comparison between parameter optimal and PID type iterative learning control with the application on a robot arm is presented in this paper. Both control schemes are based on the principle of iterative learning control which the data from the past trial is used to modify the system performance. Specifically, they both also use the optimal gain which its solution will converge in norm to zero. The control design is simple in the sense that the requirement is to track the demanding reference target with minimum error. In this framework, the desired trajectory measured by the angle of the joint is concerned. The results show that the PID type iterative learning control give better performance than the other as it generates faster convergence. The numerical experimentation suggests however that the use of PID type is capable to track the desired trajectory precisely.

KEYWORDS: Iterative Learning Control, Convergence, Parameter Optimal & PID Type

Received: Oct 30, 2016; **Accepted:** Nov 19, 2016; **Published:** Nov 21, 2016; **Paper Id.:** IJEEERDEC20164

Original Article

1. INTRODUCTION

Iterative learning control (ILC) is a technique to control the systems that perform the same task repeatedly and periodically. It is concerned with trajectory tracking control problem, where the required trajectory is repeated over a finite duration known as the trial or iteration length. This mostly applies to robot manipulators that are required to repeat a given task with high precision. Other examples are chemical batch process, automated manufacturing plant or more generally, the class of tracking systems. The general objective of iterative learning control is to use the repetitive nature of the process to gradually improve the precision. Amann *et al.*(1996) states that the basic idea of the algorithm is to suitably use information from previous trials, often in combination with appropriate current trial information, to select the current trial input to sequentially improve performance from trial to trial in the sense that the tracking error is gradually reduced. More specifically, during the execution of algorithm at the *j*th iteration, the data as control input and error are recorded. These information are used in the learning process at the (*j*+1)th iteration to refine the control input and progressively reduce the error. After a number of iteration, the appropriate control input should be obtained and this input finally generates the desired output correctly.

Since the early introduction of iterative learning control methodology by Arimoto *et al.* (1984), the general area of ILC has been progressively made by several researchers both in term of underlying theory and application. The examples for the theoretical aspect are found in Owens and Feng (2003), Owens, Chu, and Songjun (2012), and Madady (2008). Owens and Feng presents the parameter optimization through a quadratic performance index introduced as a method to establish a new control law. This new algorithm guarantees the monotonic convergence of the error to zero if the original system is a discrete-time LTI system. In Owens, Chu, and Songjun, a

multi-parameter optimal ILC algorithm that uses an approximate polynomial representation of the inverse plant is introduced. This algorithm is capable of producing the improvements of the convergence rate as the number of parameters increases. Madady purposes a proportional plus integral and derivative type ILC. This is the new technique using the principle of optimal design to determine the PID coefficients. It can guarantee the monotonic convergence under some given conditions. The examples of the application aspect are reviewed in Tayebi (2004), Lee and Lee (2007), and Sun, Hou, and Li (2013). They all are about to utilize the iterative learning control method in the application to robot manipulator, batch process system and train trajectory tracking system respectively.

In this paper, two iterative learning control schemes for the trajectory tracking problem of rigid robot manipulators are presented and compared. One is parameter optimal iterative learning control using polynomial representations of the inverse plant. The other is the PID type iterative learning control. The convergence of the tracking error to zero over the finite time interval is guaranteed when the iteration number tends to infinity. The paper is organized as follows: In section 2 the POILC using polynomial representations of inverse plant is explained in detail. The PID type iterative learning control is also described in section 3. They both are then applied to the robot manipulator and an illustrative simulation test results are given in section 4. Finally some conclusions are made in section 5.

2. POILC USING POLYNOMIAL REPRESENTATIONS OF INVERSE PLANT

The starting point of POILC using polynomial representations of inverse plant is to consider the system in the discrete model state-space form which the states, inputs and outputs are assumed to be sampled at intervals h over a time interval $[0,N]$ and the number of samples $T = N/h$ is known. The process model is written in the state-space form

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t), \qquad x(0) = x_0 \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{1}$$

where the matrices A, B, and C have an appropriate dimensions and D is a scalar. It is useful in the analysis to replace the plant model (1) by a matrix model relating a time series vector of inputs to a time series vector of outputs for each trial. To construct this matrix model, define the time series input and the time series output on the kth trial via

$$\begin{aligned} u_k &= [u_k(0), u_k(1), \ldots, u_k(N-m)]^T, \\ y_k &= [y_k(m), y_k(m+1), \ldots, y_k(N)]^T, \\ r &= [r_k(m), r_k(m+1), \ldots, r_k(N)]^T \end{aligned} \tag{2}$$

With the above definitions, the usual formula for input-output response of the system can be written in the matrix form

$$y_k = Gu_k + d_0 \tag{3}$$

where G is called the Markov parameters of the plant (1) and has the lower triangular band structure. For simplicity, it is assumed that $m = 1$. Therefore the dimension of G is $N \times N$ which can be written as

$$G = \begin{bmatrix} CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ CA^2B & CAB & CB & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N-1}B & CA^{N-2}B & CA^{N-3}B & \cdots & CB \end{bmatrix} \tag{4}$$

Furthermore, the tracking error at iteration k is defined as $e_k = r - Gu_k - d = (r - d) - Gu_k$ and hence, without loss

of generality, it is possible to replace r by $r - d$ and therefore to assume that $d = 0$ in what follows. Equivalently, it is possible to assume that $x_0 = 0$.

The repetitive nature of the problem permits the iterative modification of the input function $u(t)$ so that, as the number of iteration increases, the system learns the input function that gives perfect tracking. More precisely, the control objective is to find a recursive algorithm denoted by f in the formula

$$u_{k+1} = f(u_k, u_{k-1}, \ldots u_{k-r}, e_{k+1}, \ldots, e_{k-s}) \tag{5}$$

with the properties that, independent of the control input u_0 chosen for the first trial,

$$\lim_{k\to\infty} \|e_k\| = 0 \quad \text{and} \quad \lim_{k\to\infty} \|u_k - u^*\| = 0 \tag{6}$$

where $\|\cdot\|$ denotes any norm for the time series.

Motivation using the Cayley-Hamilton Theorem, the following approximation to the inverse ILC algorithm is presented

$$u_{k+1} = u_k + \sum_{j=1}^{M} \beta_{k+1}(j) G^{j-1} e_k, \qquad k \geq 0 \tag{7}$$

Based on a natural multi-parameter generalization of the POILC algorithm, this algorithm specifies the parameter vector β_{k+1} by solving the optimization problem

$$\beta_{k+1} = \arg\min_{\beta_{k+1}} \{J_{k+1}(\beta_{k+1}) : e_{k+1} = r - G u_{k+1} - d_0\} \tag{8}$$

where the objective function

$$J(\beta_{k+1}) = \|e_{k+1}\|^2 + \beta_{k+1}^T W_{k+1} \beta_{k+1} \tag{9}$$

and $W_{k+1} = W^T_{k+1} > 0$ is weighting matrix introduced to ensure good conditioning of the algorithm.

Using $e_k = r - Gu_k$, $k \geq 0$, the error evolution equation becomes

$$e_{k+1} = e_k + G G^{-1}(\beta_{k+1}, G) e_{k+1} \tag{10}$$

The necessary and sufficient condition for optimality is that $\partial J / \partial \beta_{k+1} = 0$, which gives the compact formula

$$\beta_{k+1} = (W_{k+1} + M^T(e_k) M(e_k))^{-1} M^T(e_k) e_k \tag{11}$$

where

$$M(e_k) = [G e_k, G^2 e_k, \ldots, G^M e_k] \tag{12}$$

This algorithm has the desirable property as the error norm is monotonicity. It also ensures the boundedness of signals and a sequential improvement in the mean square tracking error from iteration to iteration.

3. PID TYPE ITERATIVE LEARNING CONTROL

The starting point of the PID type iterative learning control is to consider the following updating law to determine the input of the system (1)

$$u_{k+1}(t) = u_k(t) + \Delta u_{k+1}(t) \tag{13}$$

where $\Delta u_{k+1}(t)$ is a term modified by using PID scheme. Hence $\Delta u_{k+1}(t)$ is chosen as follows

$$\Delta u_{k+1}(t) = k_P e_k(t+1) + k_I \sum_{m=1}^{t+1} e_k(m) + k_D (e_k(t+1) - e_k(t)) \tag{14}$$

Therefore, the relations of (13) and (14) give

$$u_{k+1}(t) = u_k(t) + k_P e_k(t+1) + k_I \sum_{m=1}^{t+1} e_k(m) + k_D (e_k(t+1) - e_k(t)) \tag{15}$$

where the error is defined as

$$e_k(t) = y_d(t) - y_k(t) \text{ for } 1 \le t \le N \tag{16}$$

k_P, k_I, and k_D are real constant coefficients which is called proportional, integral and derivative learning gains respectively.

The gains k_P, k_I, and k_D will be determined in order to satisfy

$$\lim_{k \to \infty} (y_d(t) - y_k(t)) = 0 \text{ for } t = 1, 2, \ldots, N. \tag{17}$$

The limit shown in (17) indicates that perfect tracking accuracy is ultimately achieved as the number of iterations becomes infinite.

The system (1) can be easily transformed to the following relation

$$Y(k) = GU(k) + G_0 x_0, \quad k = 0, 1, 2, \ldots \tag{18}$$

where $U(k)$ and $Y(k)$ are input and output vector respectively in iteration k as

$$\begin{aligned} U(k) &= \begin{bmatrix} u_k(0) & u_k(1) & \cdots & u_k(N-1) \end{bmatrix}^T, \\ Y(k) &= \begin{bmatrix} y_k(1) & y_k(2) & \cdots & y_k(N) \end{bmatrix}^T \end{aligned} \tag{19}$$

G_0 is the matrix, $G_0 = \begin{bmatrix} CA & CA^2 & \cdots & CA^N \end{bmatrix}^T$ and G is the same matrix as shown in (4).

The system relation shown in (6) can be rewritten in the form

$$Y(k+1) = Y(k) + GV(k) \qquad k = 0, 1, \ldots \tag{20}$$

where

$$V(k) = U(k+1) - U(k) \tag{21}$$

The error dynamic of the system can be obtained from (20) as

$$E(k+1) = E(k) - GV(k) \qquad k = 0, 1, \ldots \tag{22}$$

where $E(k)$ is the error vector at iteration k which is defined as

$$E(k) = Y_d - Y(k) = \begin{bmatrix} e_k(1) & e_k(2) & \cdots & e_k(N) \end{bmatrix}^T, \quad Y_d = \begin{bmatrix} y_d(1) & y_d(2) & \cdots & y_d(N) \end{bmatrix}^T \tag{23}$$

From the definition of $V(k)$ and $E(k)$, the control law (15) can be written in the following compact form as

$$V(k) = \{(k_P + k_D)I + k_I F_1 - k_D F_2\} E(k) \text{ for } k = 0,1,\ldots \tag{24}$$

where I is $N \times N$ identity matrix and F_1 and F_2 are defined as

$$F_1 = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix}, \quad F_2 = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \tag{25}$$

Substituting $V(k)$ from (24) into error dynamic equation (22) yields

$$E(k+1) = G_c E(k) \qquad k = 0,1,\ldots \tag{26}$$

where G_c is a matrix obtained by

$$G_c = I - (k_P + k_D)G - k_I G F_1 + k_D G F_2 \tag{27}$$

To achieve a tracking accuracy (17), it depends on matrix G_c in which the gains k_P, k_I, and k_D should be selected appropriately.

This algorithm guarantees the monotonic convergence if the sufficient condition is obtained in terms of system Markov parameters and the appropriated gains of the optimal PID approach are selected.

4. SIMULATION RESULTS

In order to have some idea about the effectiveness of these two algorithms in term of reducing tracking error and see how the comparison of the accuracy to the target of both algorithms would be, a simulation study is conducted. Let consider one-link one degree of freedom arm with revolute joint shown in Figure 1.

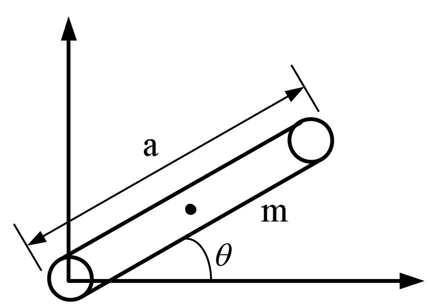

Figure 1: Configuration of a One-Link Arm

This system is described by the state space form as follows (see Saha (2008))

$$\dot{y}_1 = y_2$$
$$\dot{y}_2 = \frac{3}{m a^2}\left(\tau - \frac{1}{2} m g a \sin\theta\right) \tag{28}$$

Where the 2-dimensional state-space vector is given by $\mathbf{y} \equiv [y_1\ y_2]^T = [\theta\ d\theta/dt]^T$. Also a is the length of the link, m is the mass of the link, θ is angle of the link, and τ is the torque input to the system. This system is to determine the input torque τ applied to the link so that the angle of the link θ approaches the desired given angle θ_d in time interval $[0, T]$, By increasing the number of iterations, error between θ and θ_d is gradually reduced and becomes zero eventually. The physical parameters and desired trajectory are listed as follows.

Physical parameters: $m = 0.5$ kg, $a = 0.3$ m, $g = 9.81$ m/s^2

Desired trajectory: $\theta_d = 5\sin 0.5t$ for $t \in [0,14]$, the time interval $T = 14$

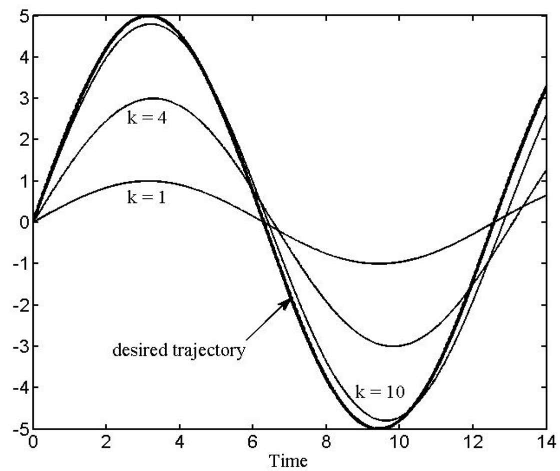

Figure 2: The Output Trajectory of Control Law (7) at the Iteration k = 1, 4 and 10

Applying the control law (7) with $M = 10$ and the desired trajectory over the time interval. A small value of $W = 10^{-6}I$ is selected to provide maximal convergence rates whilst avoiding and numerical problems in evaluating β_{k+1}. Simulation results are shown in Figure 2 and 3.

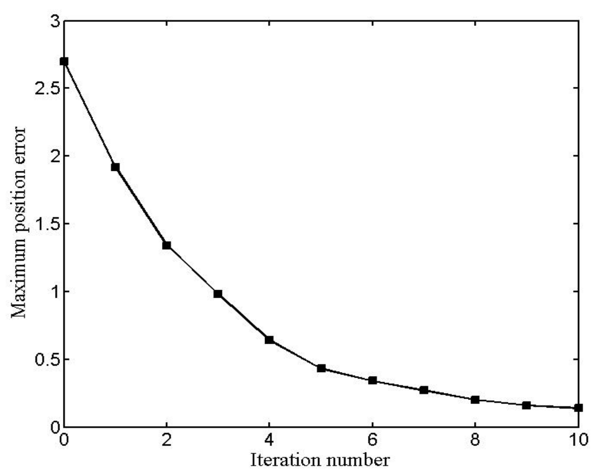

Figure 3: The Tracking Error Versus the number of Iterations for the Link of Control Law (7)

The results indicate that by increasing the iteration number, the output converges to the given desired trajectory. It is obviously shown that the convergence is monotonic as the tracking error is gradually decreased as the iteration number increases.

Again, the control law (15) is applied with the optimum selection of gains k_P, k_I, and k_D according to the procedure is shown step by step in Madady (2008). Hence the optimal coefficients k_P, k_I, and k_D is chosen as k_P, k_I, and k_D equal to -0.0877, 1.358×10^{-4}, and 43.291 respectively. The same desired trajectory is also selected with the same time interval. This leads to the results given in Figure 4 and 5.

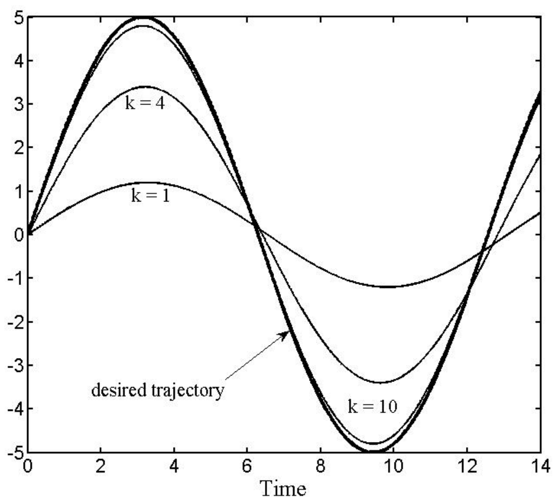

Figure 4: The Output Trajectory of Control Law (15) at the Iteration *k* = 1, 4 and 10

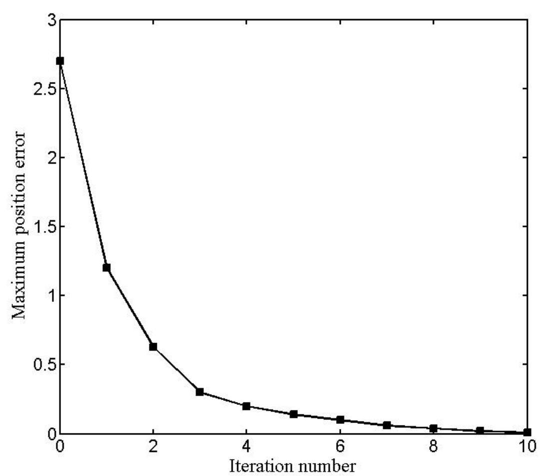

Figure 5: The Tracking Error Versus the Number of Iterations for the Link of Control Law (15)

The results obviously show that the output can track the desired trajectory as the number of iteration increases. It is also showed that the convergence in this simulation is monotonic.

In order to see how different between POILC using polynomial representations of inverse plant algorithm and PID type algorithm can track the desired trajectory precisely, the convergence of both algorithms are compared and shown in Figure 6.

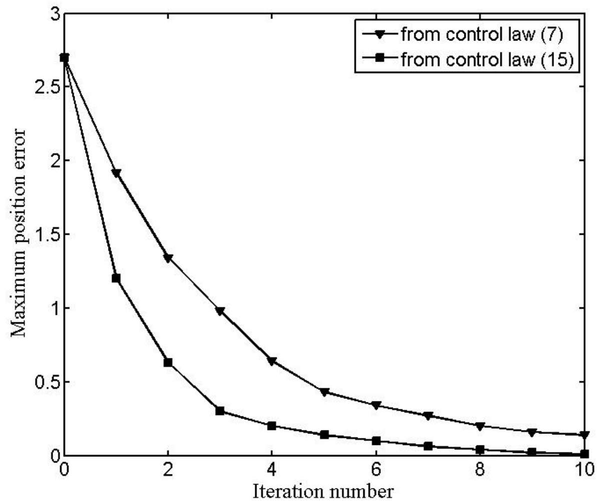

Figure 6: The Convergence Comparison between Two Algorithms

It is apparently seen in Figure 6 that both algorithm has the same trend of convergence as they are monotonic. However, it is seen that applying PID type ILC algorithm can improve the convergence more than POILC using polynomial representations of inverse plant algorithm. The angle of the link when applying the last algorithm can track the desired position trajectory more quickly and the convergence rate is much more.

5. CONCLUSIONS

In this paper two iterative learning control algorithms, POILC using polynomial representations of inverse plant and PID type ILC, have been explained and applied to the position tracking problem of one link robot manipulator. These two algorithms are based on parameter optimization which the suitable parameters are selected during each repetition in order to improve performance. Results from simulation application of these control laws to robot manipulator are presented and also show that control over performance aspects is possible. The results of both algorithms show the same behaviors. The position tracking errors monotonically decrease with the increase of the iteration number. However, the PID type ILC gives a better performance as it is efficiency to track the desired trajectory more accurately and faster under the same number of iteration. The extending to control the two-link robot manipulator with these algorithms is interesting in the future research.

6. REFERENCES

1. *Amann, N., Owens, D. H., and Rogers, E.(1996). Iterative learning control using optimal feedback and feedforward actions. International Journal of Control, vol. 65, 277 – 293.*
2. *Arimoto, S., Kawamura, S., and Miyazaki, F.(1984). Bettering operation of robots by learning. Journal of Robotics Systems, 1(2), 123 – 140.*
3. *Owens, D. H., and Feng, K.(2003). Parameter optimization in iterative learning control. International Journal of Control, 76(11), 1059 – 1069.*
4. *Owens, D. H., Chu, B., and Songjun, M.(2012). Parameter-optimal iterative learning control using polynomial representations of the inverse plant. International Journal of Control, 85(5), 533 – 544. doi: 10.1080/00207179.2012.658867.*

5. *Madady, A.(2008). PID type iterative learning control with optimal gains. International Journal of control, Automation, and Systems, 6(2), 194 – 203.*

6. *Tayebi, A.(2004). Adaptive iterative learning control for robot manipulaotrs. Automatica, 40, 1195 – 1203. doi:10.1016/j.automatica.2004.01.026.*

7. *Lee, J., and Lee, K.(2007). Iterative learning control applied to batch process: An overview. Control Engineering Practice, 15(10), 1306 – 1318.*

8. *Sun, H., Hou, Z., and Li, D.(2013). Coordinated iterative learning control schemes for train trajectory tracking with overspeed protection. IEEE transactions on Automation Science and Engineering, 10(2), 323 – 333.*

9. *Saha, S. K. Introduction to Robotics. New Delhi, Tata McGraw-Hill.*